ACE: Um Agente de Compras na Internet

Ana Beatriz Neto
Duarte Gouveia
Mário J. Silva

DI-FCUL TR-98-6

May 1998

Departamento de Informática
Faculdade de Ciências da Universidade de Lisboa
Campo Grande, 1700 Lisboa
Portugal

Technical reports are available at `http://www.di.fc.ul.pt/biblioteca/tech-reports`. The files are stored in PDF, with the report number as filename. Alternatively, reports are available by post from the above address.

ACE: Um Agente de Compras na Internet

Ana Beatriz Neto *
Duarte Gouveia *
Mário J. Silva **

* Departamento de Engenharia Electrotécnica e Computadores, Instituto Superior Técnico, Portugal
{ana.neto, duagouveia} @mail.telepac.pt

** Departamento de Informática, Faculdade de Ciências da Universidade de Lisboa, Portugal
mjs@di.fc.ul.pt

Sumário

O comércio através da Internet encontra-se em franca expansão, mas existem ainda dificuldades a ultrapassar para que qualquer utilizador se possa tornar um comprador. O problema mais discutido tem sido o da segurança das transacções, mas existem outros problemas que tornam complicado o processo da compra. Ao construirmos o ACE (Agente de Compras Especializado), propusemo-nos a resolver um desses problemas, o de encontrar o produto ou serviço que melhor corresponde às necessidades do comprador, através da criação de uma arquitectura escalável (já que se trata de software que poderia ser o elo de ligação entre todos os compradores e todos os vendedores da Internet) e de um protótipo que demonstrasse a funcionalidade dessa arquitectura. Ao contrário da maior parte da literatura existente nesta área, que apresenta apenas a funcionalidade pretendida em software deste tipo, neste artigo descrevemos detalhadamente a nossa proposta de arquitectura, baseada no paradigma dos agentes, e a metodologia utilizada para o seu desenvolvimento. Concluímos com uma análise comparativa entre o ACE e um produto comercial similar.

Palavras chave - Agentes, comércio electrónico, engenharia de *software*

Abstract

Electronic commerce through the Internet is a growing business, but there are still some problems to overcome so that every user can become a buyer. Safe transactions are the most discussed problem, but there are other problems that make shopping on the Internet difficult. When we built ACE (which stands for Agente de Compras Especializado) we aimed at solving one of those problems: finding the product or service that best suits your needs. We designed a scalable model for that purpose (since this software could be used to connect every buyer and every seller on the Internet) and we built a prototype that shows how this model works. We describe our proposed model in detail, based on the software agents paradigm, and the methodology we used to develop it. We conclude with a comparison between ACE and a similar commercial product..

Keywords - Agents, Internet shopping, spiral methodology.

1. Introdução

Presentemente, o processo pelo qual se efectuam compras na Internet de forma mais simples é muito semelhante àquele utilizado nas compras convencionais. Em primeiro lugar, para comprar um produto é necessário encontrar uma loja on-line que forneça esse produto. Uma vez encontrada a loja, é necessário procurar o produto dentro da loja. Este processo terá que ser repetido para mais lojas até que o produto seja encontrado com as melhores características possíveis, tendo como limite o tempo que o comprador está disposto a despender no processo de compra.

O primeiro problema com que se depara um potencial comprador na Internet é a ausência de *software* específico para o auxiliar nesse processo. Assim, muitas vezes esse comprador tentará encontrar o produto que procura utilizando directamente motores de busca, o que raramente lhe trará bons resultados. Por exemplo, se quem tenta comprar um livro colocar o título do livro como base da pesquisa num motor de busca, provavelmente obterá muitas respostas, mas poucas ou nenhumas lhe permitirão efectuar a aquisição do livro. Se o potencial comprador utilizar um processo de compra análogo ao convencional, procurando primeiro a loja e depois o produto dentro desta, será muito mais provável que consiga obter informação relevante, mas através de um processo demorado e repetitivo. Para além disso, a comparação entre produtos não será fácil, porque os produtos são apresentados pelas várias lojas em formatos diferentes e essa apresentação não é feita em simultâneo. Como resultado, é frequente que o comprador desista ou acabe por se contentar com algo que não é o que pretendia por não querer perder mais tempo.

Neste artigo descrevemos um modelo arquitectural e o processo de desenvolvimento do protótipo de um agente de compras que pretende encontrar soluções para melhorar a situação que acabámos de descrever. Ao utilizar esse agente, que desenvolvemos e designámos por ACE (Agente de Compras Especializado), o comprador:

- deixa de ter que se preocupar com as diferentes interfaces das lojas, já que a interface será só uma durante todo o processo de procura dos produtos;
- passa a ver os produtos e as suas características num formato que facilita a sua comparação;
- poupa tempo e dinheiro, porque o ACE, para além de ser mais rápido a fazer as procuras (já que as pode efectuar em paralelo em várias lojas), as pode efectuar sem que o utilizador esteja on-line;
- encontra mais vezes aquilo que pretende, porque o factor tempo deixará de ser limitativo e o ACE efectua sem dificuldade uma pesquisa exaustiva em todas as lojas conhecidas.

O artigo encontra-se estruturado de seguinte forma: na secção 2 começamos por apresentar o conceito de agente, em seguida, indicamos os problemas que existem hoje em dia para encontrar informação na Internet, para depois mostrarmos o papel que os agentes podem ter nesse contexto; na secção 3 descrevemos o ACE e a forma como foi desenvolvido o protótipo actual; na secção 4 apresentamos um outro agente de compras na Internet, comparando-o com o ACE; na secção 5 apresentamos as nossas conclusões e as direcções em que poderá evoluir este trabalho no futuro.

2. Agentes

2.1 O que é um agente?

Consultando um dicionário, verificamos que um agente é "Tudo o que age, que actua; o que cuida de negócios por conta alheia"[1]. No entanto, é óbvio que o significado da palavra poderá variar consoante o contexto em que se insira.

A sua inserção no contexto da informática tem-se tornado, nos últimos anos, cada vez mais popular. Mas, neste contexto, a palavra "agente" não tem uma definição consensual, talvez por cobrir diversas áreas de investigação e desenvolvimento. Para uma análise de algumas dessas definições poderá consultar [7]. Algumas das características que são associadas aos agentes nessas definições e relevantes para a nossa aplicação são:

- **Autonomia:** Um agente será tão mais autónomo, quanto mais controlo tiver sobre as suas acções. Um agente pode ser considerado autónomo em relação ao ambiente ou em relação aos outros agentes.
- **Pro-actividade:** Um agente pro-activo toma a iniciativa para atingir os seus objectivos, não se limitando a responder a estímulos do ambiente.
- **Reactividade:** O agente tem capacidade de reagir às mudanças que sente no ambiente (estímulos).
- **Continuidade temporal:** A continuidade temporal impõe que o agente esteja continuamente activo. Note-se que grande parte do *software* existente não tem esta característica, já que executa uma ou mais tarefas e termina.
- **Capacidade social:** Se um agente tem capacidade social então esse agente comunica com outros agentes, o que poderá incluir humanos. Dessa comunicação, que será feita utilizando uma linguagem de comunicação entre agentes, poderá resultar uma cooperação. Para o caso específico da comunicação entre o agente e o utilizador humano deverá ocorrer uma cooperação na construção do "contrato" sobre o que o agente deverá fazer e não uma simples ordem.
- **Capacidade de adaptação:** Um agente com capacidade de adaptação é capaz de alterar o seu comportamento com base na experiência. Assim, diz-se também que esse agente tem capacidade de aprendizagem. A adaptação pode ser relativa ao ambiente ou no sentido de melhorar a sua interacção com outros agentes, nomeadamente, com os utilizadores humanos.
- **Mobilidade:** A característica da mobilidade corresponde à capacidade do agente para se mover no ambiente. Quando se trata de agentes de *software*, um agente móvel é aquele que é capaz de se transportar de uma máquina para outra durante a sua execução.

2.2 Agentes para a recuperação de informação na Internet

Actualmente para aceder à informação presente na Internet o utilizador recorre muitas vezes a técnicas de recuperação de informação (*information retrieval*). Essas técnicas auxiliam-no a procurar o que pretende num conjunto de informação maioritariamente irrelevante. À medida

[1] Dicionário Universal da Língua Portuguesa, Texto Editora

que a quantidade de informação se torna cada vez maior, torna-se mais difícil para os utilizadores encontrarem o que procuram.

A forma mais comum para auxiliar as pesquisas na Internet são os motores de pesquisa existentes na World Wide Web (como o Yahoo!, o AltaVista, o HotBot ou o Lycos[1]). Estes motores de pesquisa são certamente uma boa forma para encontrar documentos que contenham determinadas palavras, designadas por palavras chave. No entanto, têm diversos problemas:

- Assumem que o utilizador que procura informação sabe escolher as melhores palavras chave para obter os resultados mais relevantes (ou que consegue percorrer uma hierarquia de assuntos até encontrar o que pretende);
- Não conseguem encontrar a informação que é acessível através da World Wide Web mas que é gerada dinamicamente (por CGIs, por exemplo);
- Procuram a informação num contexto genérico, o que faz com que surja como resposta muito mais informação irrelevante (por exemplo, se se pretender encontrar um filme e se fizer a pesquisa num motor de busca especializado em cinema, o número de respostas relevantes será certamente maior)
- A forma de introduzir as palavras chave e de apresentar as respostas varia de um motor de pesquisa para outro (existem motores de pesquisa, nomeadamente o MetaCrawler e Cyber411[2] que através de uma única interface, fazem a mesma pesquisa em vários motores de pesquisa numa tentativa de resolver este problema);
- É necessário um servidor poderoso para efectuar a indexação dos muitos documentos existentes, guardar o índice e suportar a pesquisa sobre esse índice por muitos utilizadores, sendo este um problema que se agrava à medida que o número de documentos aumenta;
- Os motores de pesquisa não são completos, não indexando a totalidade dos documentos existentes e ficando frequentemente com referências desactualizadas;
- Os diversos motores de pesquisa não cooperam entre si, levando a que grande parte do esforço de indexação seja repetido;
- É necessário que o utilizador mantenha uma ligação à Internet durante todo o tempo necessário para efectuar a pesquisa.

Se puderem delegar a pesquisa de informação a um agente (ou mais), os utilizadores gastarão muito menos tempo com essa tarefa. Dessa economia de tempo resultarão ainda poupanças a nível monetário, por se conseguir reduzir o tempo de ligação. Note-se que os agentes não se importam de trabalhar a qualquer hora do dia, o que poderá levar a uma melhor distribuição do tráfego ao longo do dia.

Por outro lado, os agentes poderão constituir uma melhor interface para a recuperação de informação na Internet, se existir uma colaboração na formulação do pedido, de forma a maximizar a relevância dos resultados a obter e/ou se o agente for especializado e for ele próprio a fornecer o contexto em que deve ser interpretado o pedido (por exemplo, um agente

[1] http://www.yahoo.com/ , http://www.altavista.com/ , http://www.hotbot.com/ , http://www.lycos.com/
[2] http://www.metacrawler.com/ , http://www.cyber411.com/

para fazer pesquisas sobre filmes, introduzindo apenas o título do filme pretendido, sabe sempre que aquilo que foi introduzido deverá corresponder a um título de um filme).

Se considerarmos agentes com capacidade de comunicação entre si, existe ainda a possibilidade de colaboração entre agentes, o que minimizaria o esforço individual de cada um. Ao fim ao cabo, ao dividir uma tarefa por vários agentes, há vantagens a nível do desempenho e a nível da forma como a especialização de cada agente é aproveitada.

Finalmente, os agentes poderão ser construídos de forma a obterem informação gerada dinamicamente.

Mas quem pretenda construir um agente para recuperação de informação da Internet terá que responder a algumas questões. Consideremos o caso particular da recuperação de informação na WWW:

- **Capacidade:** Até que ponto pode um agente compreender a informação presente nas páginas da Web? As anotações HTML estruturam a aparência das páginas mas não dão qualquer indicação em relação ao seu conteúdo.
- **Utilidade:** Será que aquilo que o agente consegue compreender é suficiente para auxiliar verdadeiramente os utilizadores? Será que o utilizador conseguia desempenhar a tarefa do agente tão bem ou melhor do que ele com utilizando os motores de pesquisa usuais?
- **Escalabilidade:** Será que para o agente conseguir cobrir uma parte significativa da Web é necessário um esforço não comportável por parte de quem desenvolve o agente? Será que o agente extrair automaticamente informação de páginas da Web que não conhece?
- **Mobilidade:** Será que o agente está dependente de uma máquina ou de uma plataforma?
- **Constrangimentos ambientais:** Que aspectos da Web permitem ao agente desempenhar a sua tarefa? Qual é a probabilidade de esses aspectos se manterem constantes?

3. ACE – Agente de Compras Especializado

O ACE (Agente de Compras Especializado) foi por nós desenvolvido durante o ano lectivo de 1996/97. Nesta secção começamos por expor os objectivos e funcionalidade pretendidas para o ACE, para, em seguida, apresentarmos a sua arquitectura. Finalmente, analisamos os aspectos que fazem do ACE um agente.

3.1 Objectivos e Funcionalidade

O objectivo do ACE é tornar as compras através da Internet mais fáceis para o utilizador. Isso é conseguido melhorando, especificamente, a fase do processo de compra em que o utilizador tenta obter informação sobre os produtos ou gama de produtos já seleccionados, por forma a decidir qual a melhor alternativa para a satisfação da sua necessidade.

O ACE tenta encontrar os produtos pretendidos em paralelo nas várias lojas na Internet que lhe estão associadas. Em seguida, apresenta a informação obtida de uma forma uniformizada e organizada. Perante esta informação, é muito mais fácil para o utilizador escolher o produto com as características mais satisfatórias.

Com o agente de compras, o utilizador deixa de ter que repetir o processo de procura para cada uma das várias lojas que comercializam o produto ou serviço pretendido. Para além disso, uma vez que o processo de procura utilizando o agente de compras é substancialmente mais

rápido, o tempo disponível para pesquisar informação deixa de ser um factor limitativo e o utilizador passa a encontrar mais vezes o produto que quer, ou seja, o que tem as melhores características para satisfazer a sua necessidade.

Para se alcançar este objectivo, o funcionamento do agente de compras pretendido é o seguinte:

- O utilizador escolhe o tipo de produto, de entre a lista de produtos que actualmente são disponibilizados pelo agente;
- O utilizador especifica, se o desejar, uma série de características para esse produto e algumas restrições às condições de procura;
- O agente de compras procura, dentro de produtos desse tipo, aqueles com as características desejadas (a procura pode ser efectuada sem o utilizador estar on-line, sendo os resultados visualizados mais tarde);
- Como resultado da pesquisa, o agente de compras produz uma lista de produtos e respectivas características;
- O utilizador pode pedir a ordenação da lista produzida (utilizando como critério qualquer uma das características do produto), de forma a facilitar a comparação entre os resultados obtidos.

Na Figura 1 encontra-se o écran onde o utilizador especifica as características do produto e recebe os resultados, quando se trata de livros. Como se vê na figura, existe um botão que permite parar a pesquisa e um outro para indicar que deseja consultar mais tarde os resultados. Nesse último caso, o agente continua a obter resultados enquanto o utilizador não está ligado.

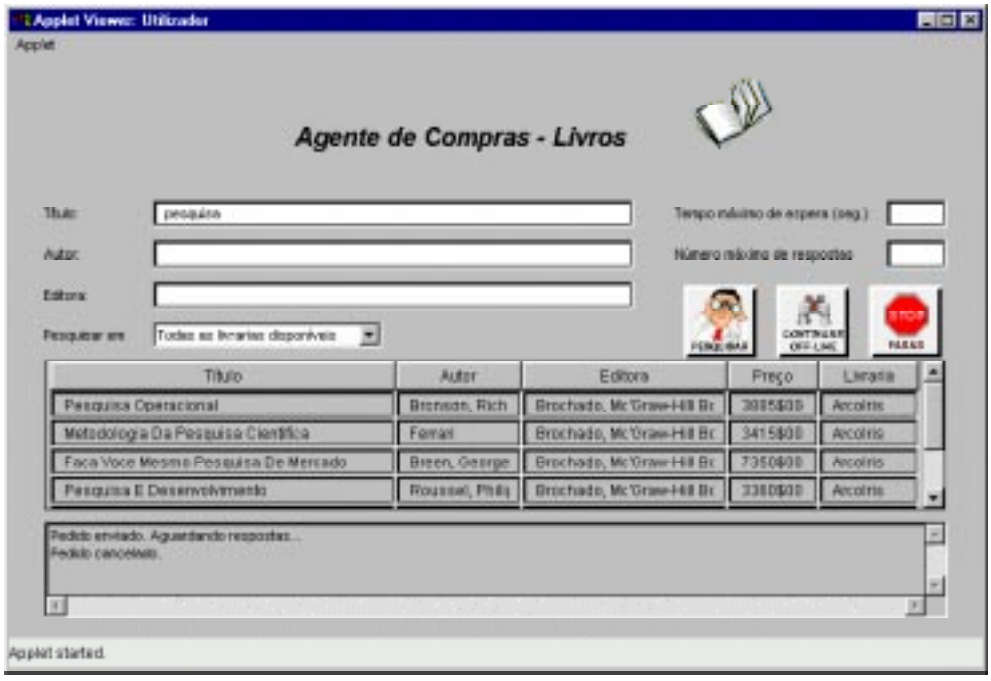

Figura 1 – Écran para o produto "Livros" depois de cancelada uma pesquisa para a qual já tinham sido obtidos alguns resultados

3.2 Arquitectura

Para que o agente de compras tenha a funcionalidade descrita e possa, no futuro, ser estendido para muitos produtos e muitas lojas desenvolvemos uma arquitectura composta por vários subagentes: *utilizador*, *porta de acesso*, *intermediário* e *facilitador*. Representamos as relações entre estes subagentes na Figura 2. Os vários subagentes comunicam entre si utilizando uma linguagem de comunicação entre agentes, o KQML[1].

O subagente *utilizador* serve como interface entre o utilizador humano e os restantes subagentes que compõem o agente de compras. Para construir a interface com o utilizador humano foi utilizada uma *applet* Java.

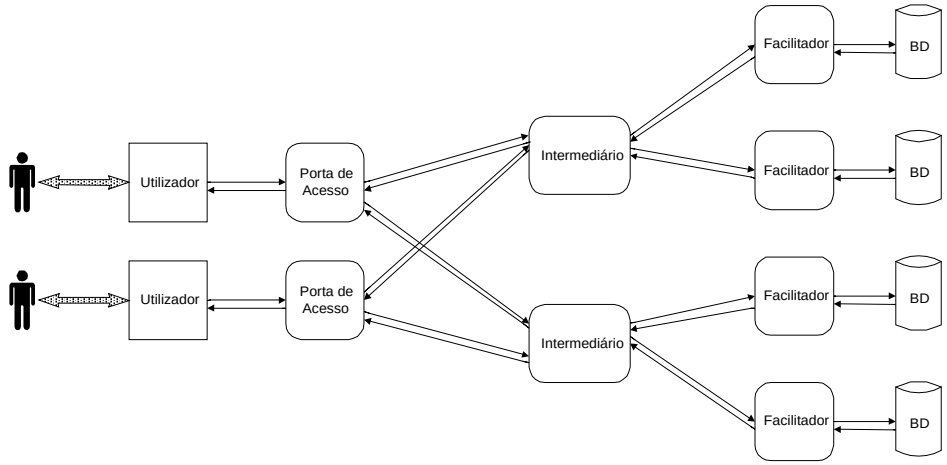

Figura 2 - Relações entre os vários subagentes que compõem o agente de compras

O subagente *porta de acesso* surgiu da necessidade de superar uma restrição de segurança dos browsers Internet actuais, que não permitem a criação de canais de comunicação para outras máquinas que não sejam nem a máquina onde a *applet* se está a executar, nem a máquina de onde foi carregada inicialmente. O subagente *porta de acesso* executa-se na mesma máquina em que se encontra a página de onde a *applet* foi carregada. A *applet* pode comunicar com o subagente distribuídos por várias máquinas na Internet porque este se encontra nessa máquina e o subagente *porta de acesso* pode comunicar com os *intermediários* (que se podem encontrar distribuídos por várias máquinas na Internet) porque já não se trata de uma *applet*. Assim, o subagente *porta de acesso* funciona exactamente como porta de acesso ao restante conjunto de agentes.

O subagente *intermediário* serve para agregar os resultados provenientes de vários *facilitadores* que têm algumas características comuns, por exemplo o tipo de produto que vendem. A existência dos *intermediários* evita que se contactem todos os *facilitadores* que existem quando isso não é necessário. Desta forma, reduz-se o número de mensagens a transitar sobre a rede e o número de pedidos que os *facilitadores* têm que tratar.

[1] O Knowledge Query and Manipulation Language é um protocolo e uma linguagem para comunicação entre agentes que pretende ser simultaneamente simples e expressivo (para mais informações consultar [5])

O subagente *facilitador* funciona como interface entre as lojas virtuais na Web e os restantes subagentes do ACE. O *facilitador* contacta a loja através das suas páginas na Internet, efectua o pedido e interpreta as páginas com as respostas geradas pela base dados da loja, convertendo-as para o formato que é entendido pelos restantes subagentes.

As relações que se estabelecem entre os subagentes descritos na secção anterior e a duas entidades externas com que comunicam (os utilizadores humanos e as lojas na Web) são descritas pelo modelo de objectos[1] (Figura 3). Nessa figura vemos ainda os componentes de cada um dos subagentes.

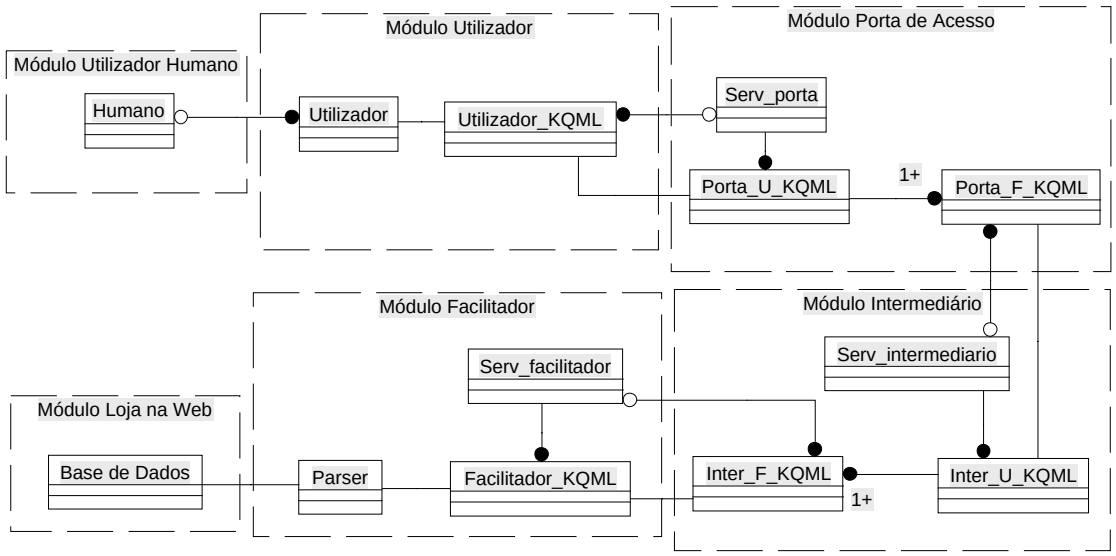

Figura 3 - Subagentes do ACE

A principal missão do ACE é satisfazer a entidade externa utilizador humano. O subagente *utilizador* tentará proporcionar a interface que melhor se adeque a realização dos pedidos do utilizador humano. Um utilizador humano pode possuir várias instâncias do subagente *utilizador* a trabalhar para ele em simultâneo. Cada instância do subagente *utilizador* pode estar quando muito ligada a um utilizador humano.

O subagente *utilizador* é a única parte do ACE que é móvel. Este subagente é constituído por uma interface (Utilizador) e por um módulo de comunicação (Utilizador_KQML).

O subagente *porta de acesso* funciona como encaminhador de pedidos a *intermediários* que podem estar localizados noutras máquinas na Web. Este subagente é constituído por um servidor que atribui a cada pedido um servidor dedicado (Porta_U_KQML). O Porta_U_KQML é responsável pela comunicação KQML através do socket que partilha com o Utilizador_KQML. Para cada canal de comunicação que for estabelecido com outros subagentes, o Porta_U_KQML criará um Porta_F_KQML para tratar da comunicação com esse subagente. A escolha dos nomes destas componentes deve-se, em primeiro lugar, ao nome do subagente (porta), ao sentido em que estão a tratar a comunicação (U para o sentido do *utilizador* e F para o sentido do *facilitador*) e, finalmente, a indicação de que trata de um canal de comunicação (KQML).

[1] O modelo de objectos do OMT corresponde ao diagrama de classes do UML (ver [6] e [11]).

O subagente *intermediário* tem a função de barrar os pedidos que não digam respeito aos *facilitadores* que lhe estão associados. Assim, um *intermediário* que só tem associados vendedores de livros pode recusar qualquer pedido de outro produto, porque sabe quais os produtos que são vendidos pelos *facilitadores* seus associados. O subagente *intermediário* tem uma estrutura com um servidor em tudo idêntica à descrita para o subagente *porta de acesso*.

O subagente *facilitador* é o responsável pela comunicação com a loja na Web e tem a capacidade de transformar as perguntas num formato que a loja na Web entende e sabe responder. Posteriormente, o *facilitador* tem a responsabilidade de interpretar as respostas produzidas pela loja e transformá-las num formato compreensível para os restantes subagentes deste sistema.

O subagente *facilitador* é constituído por um servidor que para cada pedido cria um Parser e um Facilitador_KQML. O servidor aceita pedidos de ligação da mesma forma que o servidor do *intermediário* ou da *porta de acesso*. O Parser tem a responsabilidade de comunicar com a loja na Web e o Facilitador_KQML comunica com Inter_F_KQML do subagente *intermediário*.

A entidade externa loja na Web tem a intenção de vender os seus produtos ao utilizador humano, mas as lojas na Web devem prestar informações verdadeiras e com o formato esperado para que o nosso agente de compras funcione correctamente. Se uma loja não estiver a fornecer informações durante um determinado período, será ignorada pelo agente de compras durante esse período.

Para que se possa compreender qual o comportamento do ACE, apresentamos em seguida cenários típicos de sequências de interacção que ajudam à compreensão dos fluxos de informação necessários para a realização das operações usuais.

Pergunta simples

Uma pergunta simples consiste numa pergunta à qual existem uma certa quantidade de respostas associadas e que termina correctamente.

A sequência inicia-se com a pergunta do utilizador humano à base de dados. O subagente *utilizador*, antes de enviar a pergunta, envia o pedido de identificação à *porta de acesso*, que por sua vez envia a mensagem a todos os *intermediários* e estes aos *facilitadores*. Caso os *facilitadores* associados a algum *intermediário* não vendessem o produto indicado no pedido então os *intermediários* responderiam imediatamente ao *porta de acesso* indicando a sua indisponibilidade para tratarem daquele pedido. Nessa situação os *facilitadores* nem chegariam a receber o pedido de identificação.

Quando os *facilitadores* recebem um pedido de identificação, respondem imediatamente sem terem necessidade de consultarem a loja na Web. Essas respostas são enviadas até ao *utilizador* que as guarda.

Entretanto, o *utilizador* pode já ter enviado a pergunta propriamente dita. A pergunta é propagada até ao *facilitador* que a transforma num acesso à loja na Web. À medida que são obtidas respostas, estas são enviadas até ao utilizador humano. O *utilizador* determina em que loja na Web foi produzida a resposta e junta a informação da loja nos dados a mostrar ao utilizador humano.

Quando não existem mais respostas, é emitida uma mensagem que indica isso mesmo. Essa mensagem é propagada até ao *utilizador* e termina a execução do pedido.

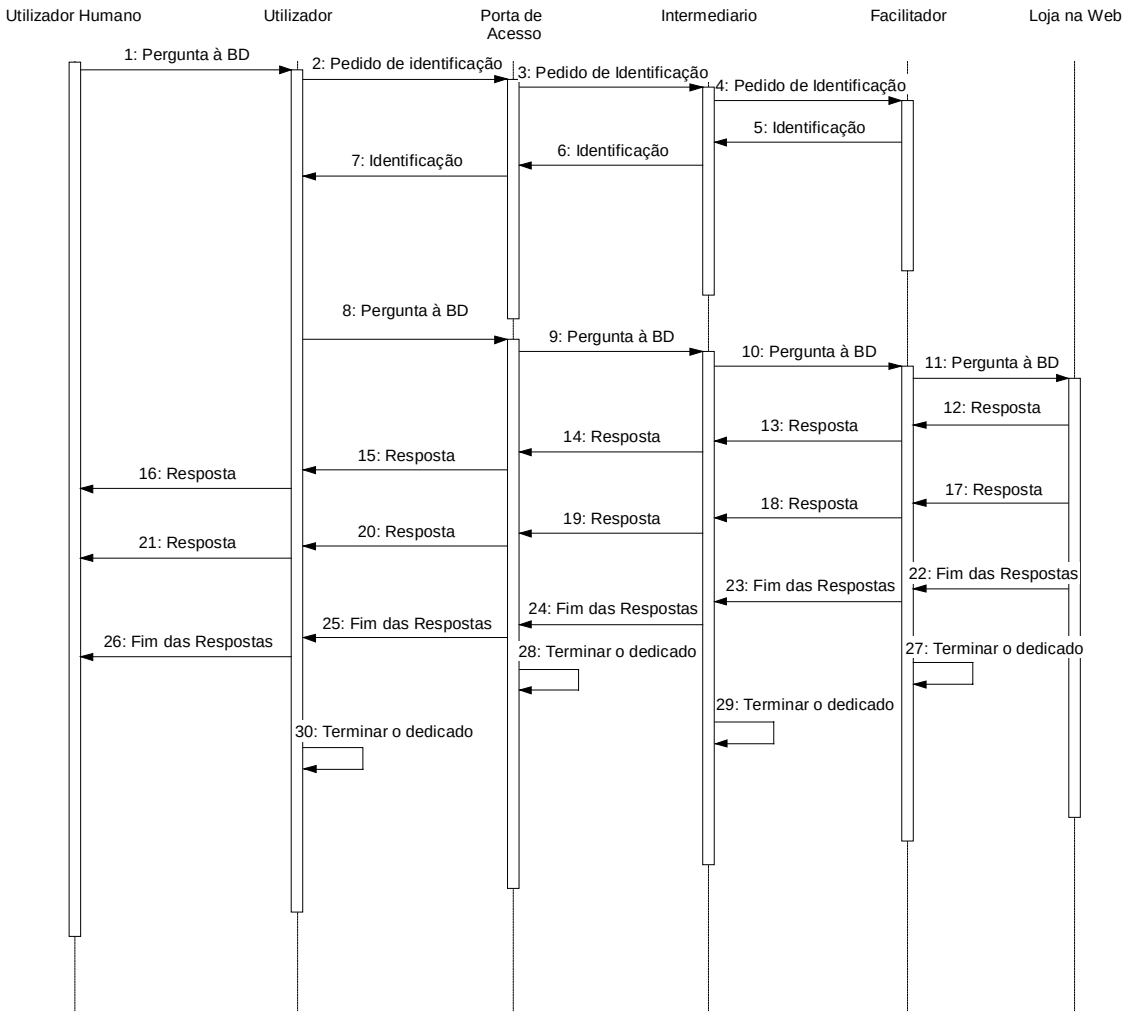

Figura 4 - Cenário da pergunta simples

Pergunta simples com cancelamento

A pergunta simples com cancelamento é o cenário em que é efectuada uma pergunta e antes de ter obtido todas as respostas o utilizador humano cancela o pedido efectuado.

A parte inicial do envio da pergunta é exactamente igual à descrita na secção 0 para a pergunta simples. Quando o utilizador humano interrompe a pesquisa com o cancelamento então o pedido de cancelamento é propagado até aos *facilitadores*, terminando em seguida a execução. Note-se que enquanto a mensagem não chega aos subagentes eles continuam a desempenhar a sua tarefa normalmente.

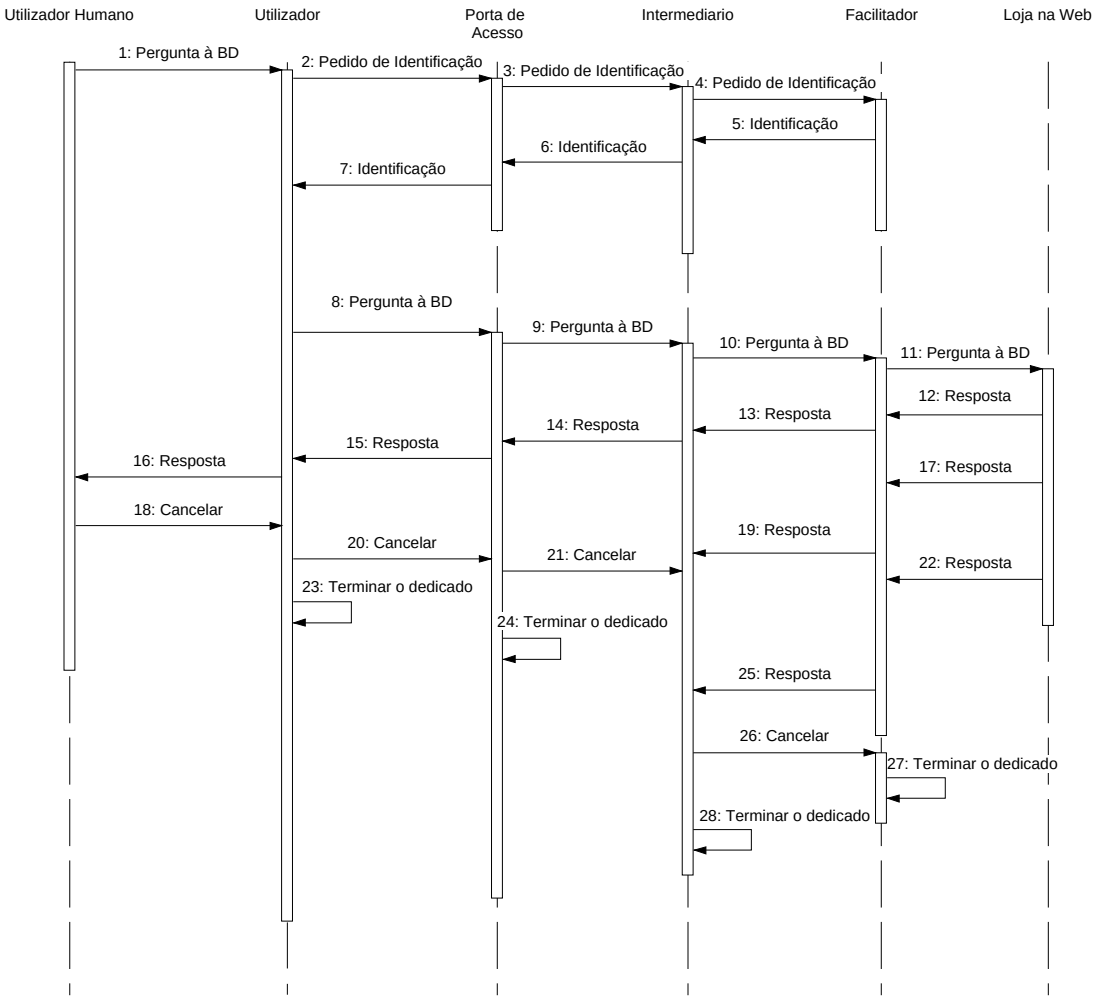

Figura 5 - Cenário da pergunta simples com cancelamento

Máquina do intermediário vai abaixo

Este cenário representa a situação em que um pedido foi efectuado e estamos a obter respostas até que, em determinado momento, a máquina em que estava instalado o intermediário deixa de funcionar. Neste cenário estamos a supor que os restantes subagentes não se encontravam na mesma máquina.

Nessa situação, os subagentes *porta de acesso* e os vários *facilitadores* que tinham canais de comunicação com o intermediário em causa vão detectar o erro no canal. O *porta de acesso* deve enviar para o *utilizador* todas as mensagens que já tinha recebido e que ainda não tinham sido tratadas. Deve ainda proceder como se todos os *facilitadores* associados a esse intermediário já tivessem retornado todas as respostas. Cada *facilitador* deve proceder como se tivesse recebido um pedido de cancelamento e terminar imediatamente a execução.

Note-se que numa pergunta à base de dados podem estar a ser consultados vários *intermediários*. Nesse caso, apenas o *intermediário* que estava na máquina que foi abaixo e os *facilitadores* que lhe estavam associados terminam, uma vez que o *porta de acesso* ao aperceber-se que existe um outro *intermediário* que mantém a pesquisa activa não interrompe a operação.

Os cenários que podem ser produzidos por máquinas irem abaixo são imensos, no entanto o comportamento dos subagentes é bastante semelhante ao que acabamos de descrever.

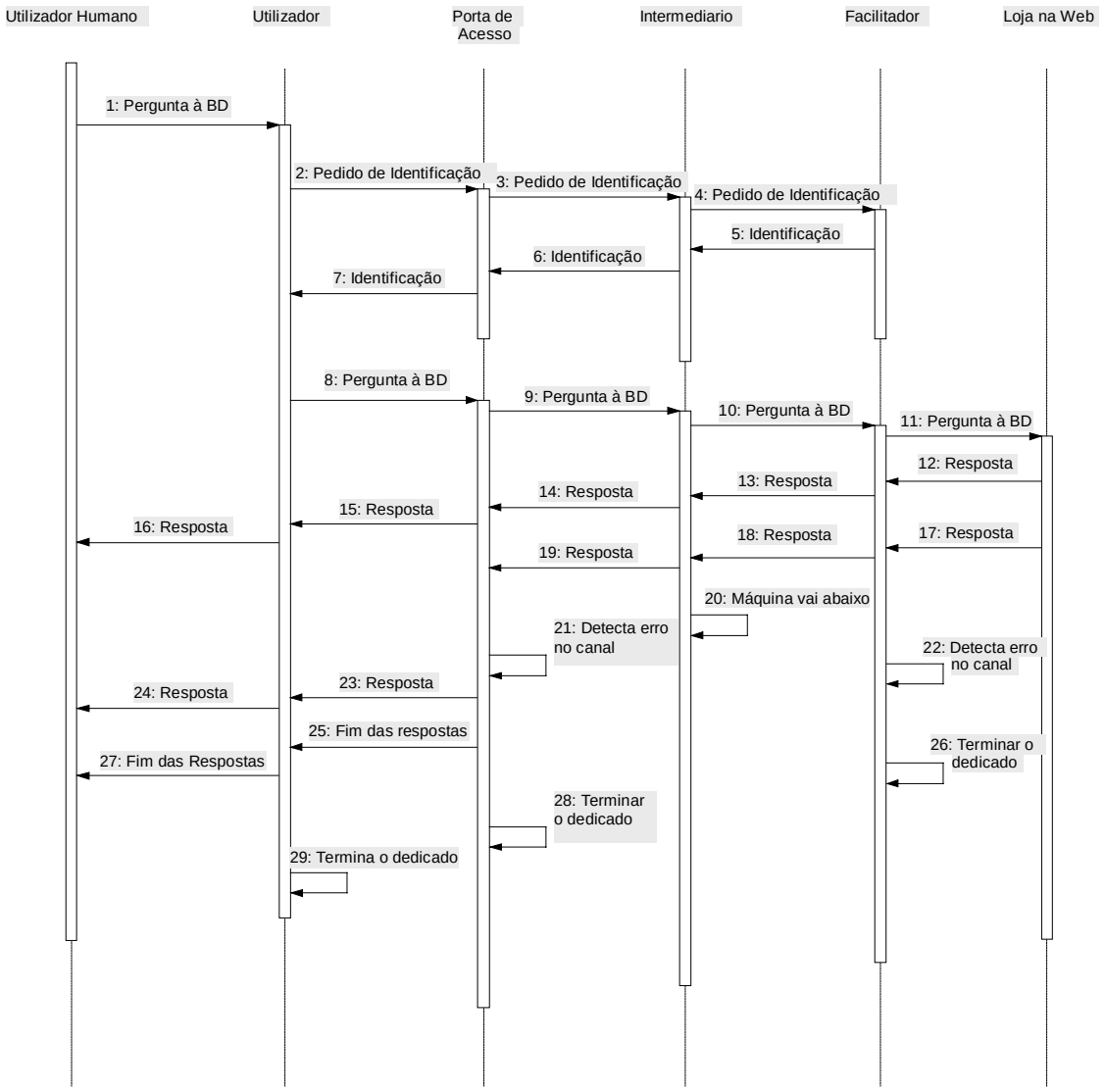

Figura 6 - Cenário em que a máquina do *intermediário* vai abaixo

Com a arquitectura que acabamos de descrever garantimos a escalabilidade do ACE, tornando-o capaz de ter associado um qualquer número de lojas e um qualquer número de utilizadores humanos sem que existam estrangulamentos (*bottlenecks*) em parte alguma do sistema de agentes.

Os subagentes *porta de acesso* não são um ponto de estrangulamento porque existe um para cada *utilizador*. Os subagentes *porta de acesso* comunicam com os *intermediários* pretendidos, para depois transferirem a informação obtida para o *utilizador* que a pediu.

Os *intermediários* agrupam conjuntos de *facilitadores* segundo critérios bem determinados, como por exemplo a localização geográfica ou o produto a vender. Assim, poderia existir um *intermediário* da Associação dos Comerciantes de Lisboa que agregaria todas as lojas de Lisboa associadas ao ACE. Caso o número de lojas deste *intermediário* fosse muito elevado, então poderia ser criado um outro nível de *intermediários* um dos quais poderia corresponder, por exemplo, às sapatarias de Lisboa. O *intermediário* da Associação dos Comerciantes de Lisboa

seria o que estaria mais perto do utilizador humano, só contactando o *intermediário* das sapatarias (que se encontra mais perto das lojas) quando fosse caso disso.

Desta forma, existe uma hierarquia de *intermediários* que se pode expandir de forma a se adaptar às necessidades reais dos utilizadores. Se essa estrutura hierárquica dos *intermediários* for, de facto, a mais adequada, não existirá também qualquer estrangulamento a esse nível.

O nível dos *faciltadores* é o que pode levantar mais problemas. Será sempre necessário existir um *facilitador* específico para cada loja, uma vez que necessitamos que exista um interprete da linguagem utilizada na loja para a linguagem do ACE. O número de *facilitadores* aumenta, portanto, de uma forma linear com um número de lojas, o que pode tornar-se problemático se se pretender abarcar toda a World Wide Web. Uma atenuante para este efeito é o facto do esforço para o desenvolvimento de cada *facilitador* ir diminuindo à medida que se avança na curva da experiência, por estes terem sempre pontos comuns.

Obviamente que este problema se resolveria se todas as lojas passassem a seguir uma norma na sua construção, por exemplo, a norma Z39.50, já utilizada no acesso a dados em bibliotecas online. Nesse caso seria muito fácil criar automaticamente *facilitadores* para os diversos tipos de lojas e um único *facilitador* poderia mesmo ser utilizado para mais do que uma loja. Na secção 4 poderemos ver que uma abordagem deste foi utilizada num outro agente de compras, mesmo sem a garantia de regularidade entre as lojas que seria dada pela existência de uma norma. Nessa secção são também explorados os problemas que foram encontrados nesse caso.

3.3 O ACE como agente

Da série de características que costumam ser relacionadas com o conceito de agente e que enumerámos na secção 2.1, pensamos que o ACE possui suficientes para ser considerado agente de acordo com diversas definições. Analisemos pois as características do nosso agente de compras:

- **Autonomia:** O nosso agente de compras é autónomo no sentido em que não recebe do utilizador qualquer indicação sobre a forma de satisfazer o pedido que lhe é formulado. Por outro lado, autores como Do, March, Rich e Wolff defendem, em [2], que um agente na Internet, para ser autónomo deve ser capaz de agir sem o utilizador esteja ligado, o que sucede com o nosso agente. Finalmente, os subagentes de que é composto o ACE, possuem também alguma autonomia, no sentido que lhe é dado por Petrie em [8], já que comunicam entre si sem ser num esquema simples de pergunta / resposta.
- **Continuidade temporal:** Aparte eventuais falhas, o ACE está continuamente activo. A única excepção é o subagente responsável pela interface com o utilizador que só está activo a partir do momento em que é lançado.
- **Reactividade:** O ACE reage a diversos estímulos do ambiente, nomeadamente, da parte do utilizador e das lojas on-line. Por exemplo, se o servidor de uma loja não estiver operacional, o agente de compras desiste de a contactar para a compra em curso. Considerando o nível dos subagentes, temos ainda que eles reagem aos estímulos dos subagentes dos níveis adjacentes. Usando um exemplo semelhante, se um *intermediário* verificar que um *facilitador* não responde desiste de o contactar para a compra em curso.
- **Capacidade social:** O agente de compras, no seu todo, não tem capacidade social. No entanto, os diversos subagentes que o compõem têm essa capacidade, comunicando entre si em KQML e cooperando para atingir um objectivo comum.

- **Capacidade de adaptação:** A capacidade de adaptação do agente de compras faz mais sentido se a considerarmos a nível dos subagentes, se bem que tenha implicações no âmbito mais geral. Os subagentes sabem adaptar-se à existência de novos subagentes no nível seguinte e sabem também adaptar-se ao desaparecimento (temporário) de subagentes com que comunicam usualmente.
- **Mobilidade:** Apenas o subagente *utilizador*, encarregue da interface com o utilizador humano, pode ser considerado móvel (e mesmo assim de uma forma limitada), uma vez que recorre a uma *applet* Java, que é carregada de uma máquina para a máquina desse utilizador. No entanto, esta característica não é uma característica essencial do ACE.

3.4 Metodologia

Para o processo de desenvolvimento do ACE optámos pela metodologia espiral, tal como é descrita por Boehm em [1]. No início da primeira volta à espiral realizámos uma análise do sistema pretendido e efectuámos um planeamento geral. Aí optámos por ter uma versão operacional do ACE no final de cada volta à espiral, ou seja por efectuar uma prototipagem em cada volta. Em cada prototipagem procurámos seguir o clássico modelo cascata.

Esta escolha permitiu-nos lidar com as incertezas inerentes ao facto de nunca termos realizado nenhum projecto deste tipo anteriormente e ao facto da Internet ser um meio em constante mudança. Assim, pudemos incorporar análises de riscos e revisões de planeamento, sem que, no entanto, a nossa abordagem deixasse de ser estruturada e sistemática.

No total, foram efectuadas cinco voltas à espiral. A primeira volta foi a mais prolongada, tendo durado nove semanas. A duração das restantes voltas variou entre as três semanas e meia e as seis semanas. No final de cada volta foi sempre efectuada uma análise de riscos. Só assim nos foi possível manter controlo sobre o projecto. Só na última volta sentimos a necessidade de uma especificação mais formal, que permitisse a consolidação do modelo arquitectural que estávamos a desenvolver. Optámos por utilizar a técnica OMT (Object Modelling Technique)[1].

4. Jango - Outro agente de compras na Internet

É possível encontrar na Internet alguns agentes de compras com objectivos semelhantes aos do ACE, mas com diversos graus de complexidade. O caso mais interessante é o do Jango (http://www.jango.com), projecto comercial que surgiu a partir de um projecto universitário, o ShopBot (para mais informações poderá consultar [4]).

A parte mais interessante do Jango é o facto de seguir uma abordagem de criação automática dos "adaptadores de informação", que correspondem aos *facilitadores* do ACE. Para descobrir como efectuar o *parsing* das páginas de resposta das diversas lojas, o Jango explora diversas regularidades encontradas entre as lojas na World Wide Web, nomeadamente:

- As lojas on-line são pensadas de forma a que os utilizadores encontrem os produtos rapidamente, por isso grande parte delas possui um página que permite efectuar procuras;
- As lojas procuram ter uma certa uniformidade em termos de *look & feel* que faz com que os produtos de uma loja sejam tipicamente todos descritos num mesmo formato;

[1] Consultar [11]

- As lojas usam o espaço em branco para facilitar a compreensão dos seus catálogos. Assim, a descrição de um produto tipicamente começa numa nova linha.

No entanto, a obtenção das funções de *parsing* é complicada porque:

- Existe muita informação irrelevante (figuras, publicidade, ligações para outras páginas, etc.);
- Nem todas as descrições de produtos contém o nome do produto e este, por vezes, surge em zonas da página que não têm nada a ver com a descrição do produto;
- Nem todas as descrições de produtos contém um preço (por vezes é preciso consultar uma outra página, outras vezes essa informação simplesmente não é fornecida);
- Nem todas as lojas obedecem às regularidades.

Assim, é natural que muitas vezes os adaptadores criados automaticamente não funcionem adequadamente, sendo necessário corrigir o seu código manualmente. Por outro lado, o facto de a estrutura das páginas de resposta mudar com alguma frequência (sobretudo devido a mudanças no aspecto das páginas), faz com que seja imprescindível a existência de código de controlo que permita verificar que um adaptador deixou de produzir dados úteis e que os seus resultados devem ser ignorados. Essa é também uma tarefa problemática.

5. Conclusões e hipóteses de trabalho futuro

Pensamos que com este trabalho conseguimos desenvolver um protótipo que demonstra que as compras na Internet podem ser significativamente facilitadas para o utilizador se lhe for oferecido um agente:

- com uma interface única e que auxilie a comparação entre os produtos;
- com capacidade de reduzir drasticamente o tempo necessário para obter resultados;
- que possa efectuar as pesquisas off-line, mostrando os resultados mais tarde.

Todas as pessoas que experimentaram o protótipo manifestaram-se satisfeitas com o aumento de rapidez da procura e com a facilidade de utilização.

Conseguimos ainda desenvolver uma arquitectura escalável, que poderia cobrir todas as lojas da Internet. O facto de o agente de compras aceitar, de forma automática, novos *facilitadores* que sejam introduzidos dinamicamente é também uma vantagem. Finalmente, é ainda uma vantagem o facto de o ACE ser compatível entre plataformas, graças à utilização da linguagem Java.

No entanto, é claro que as potencialidades do agente de compras não se esgotaram com o nosso trabalho e que o protótipo apresentado poderia ainda evoluir em várias direcções. Em primeiro lugar, poderia cobrir outros aspectos da compra, nomeadamente:

- Ajudando o utilizador a decidir que produto comprar;
- Ajudando o utilizador a encontrar opiniões sobre o produto (de especialistas, de associações de defesa do consumidor, etc.);
- Dando sugestões, com base no seu conhecimento do utilizador;
- Descobrindo novidades, descontos e preços especiais;

- Efectuando a compra, ou apenas o pagamento, nos casos em que a loja o possibilitasse.

Poderia ainda trabalhar com mais produtos e mais lojas. A principal dificuldade é ser necessário criar um novo subagente *facilitador* para cada nova loja que se pretenda tratar ou que tenha sido alterada. A solução ideal seria ter um gerador automático de *facilitadores*, que criasse os *facilitadores* analisando a loja on-line[1]. Note-se, no entanto, que um gerador deste tipo nunca poderá resolver este problema por completo. É natural que, para criar *facilitadores* para lojas sem páginas de procura ou directamente para bases de dados fosse necessário que os *facilitadores* fossem criados de raiz para cada loja e não automaticamente.

6. Referências

[1] Barry W. Boehm; *A Spiral Model of Software Development and Enhancement*; IEEE Computer; 1988

[2] Orlantha Do, Eric March, Jennifer Rich, Tara Wolff; *Intelligent Agents & The Internet: Effects on Electronic Commerce and Marketing*; http://bold.coba.unr.edu/Tara/paper.html%20copy

[3] Robert Doorenbos, Oren Etzioni, Daniel Weld; *A Scalable Comparison-Shopping Agent for the World-Wide Web*; Department of Computer Science and Engineering, University of Washington, 1996 (disponível por ftp em ftp://ftp.cs.washington.edu/pub/etzioni/softbots/agents97.ps)

[4] Stan Franklin, Art Graesser; *Is it an agent, or just a program?: A Taxonomy for Autonomous agents*; Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, Springer-Verlag, 1996 (também disponível em http://zebra.msci.memphis.edu/~franklin/AgentProg.html)

[5] Tim Finin, Rich Fritzon, Don McKay, Robin McEntire; *KQML - A Language and Protocol for Knowledge and Information Exchange*; http://www.cs.umbc.edu/kqml/papers/kbkshtml/kbks.html

[6] Martin Fowler; *UML Distilled - Applying the Standard Object Modeling Language;* Addison-Wesley; 1997;

[7] Ana Beatriz Neto, Duarte Gouveia; *Agente de Compras – Relatório Final*; http://asterix.ist.utl.pt/~agcompras/

[8] Charles J. Petrie; *Agent-Based Engineering, the Web, and Intelligence*; http://cdr.stanford.edu/NextLink/Expert.html

[9] Roger S. Pressman; *Software Engineering: a practitioner's approach*; 3rd edition (European adaptation); McGraw-Hill International; 1994

[10] RFC 1729; *Using the Z39.50 Information Retrieval Protocol in the Internet Environment*;

[11] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, W. Lorensen; *Object-Oriented Modeling and Design*; Prenctice Hall; 1991

[1] Supondo que é difícil convencer todas as lojas on-line a aceitar um formato standard, como o Z39.50